Joint Attention for Automated Video Editing

Hui-Yin Wu, Trevor Santarra, Michael Leece, Rolando Vargas, Arnav Jhala

Joint Attention for Automated Video Editing

Hui-Yin Wu
hui-yin.wu@inria.fr
Université Côte d'Azur, Inria
Sophia-Antipolis, France

Trevor Santarra
trevor@Unity3d.com
Unity Technologies
San Francisco, CA, USA

Michael Leece
michael.o.leece@gmail.com
University of California Santa Cruz
Santa Cruz, CA, USA

Rolando Vargas
rvargas1@ucsc.edu
University of California Santa Cruz
Santa Cruz, CA, USA

Arnav Jhala
ahjhala@ncsu.edu
North Carolina State University
Raleigh, NC, USA

ABSTRACT

Joint attention refers to the shared focal points of attention for occupants in a space. In this work, we introduce a computational definition of joint attention for the automated editing of meetings in multi-camera environments from the AMI corpus. Using extracted head pose and individual headset amplitude as features, we developed three editing methods: (1) a naive audio-based method that selects the camera using only the headset input, (2) a rule-based edit that selects cameras at a fixed pacing using pose data, and (3) an editing algorithm using LSTM (Long-short term memory) learned joint-attention from both pose and audio data, trained on expert edits. The methods are evaluated qualitatively against the human edit, and quantitatively in a user study with 22 participants. Results indicate that LSTM-trained joint attention produces edits that are comparable to the expert edit, offering a wider range of camera views than audio, while being more generalizable as compared to rule-based methods.

CCS CONCEPTS

• **Human-centered computing** → *Interaction design process and methods*; *User models*; • **Applied computing** → *Media arts*; • **Computing methodologies** → *Activity recognition and understanding*; *Video summarization*; *Neural networks.*

KEYWORDS

smart conferencing, automated video editing, joint attention, LSTM

IMX '20, June 17–19, 2020, Cornella, Barcelona, Spain

ACM ISBN 978-1-4503-7976-2/20/06.
https://doi.org/10.1145/3391614.3393656

ACM Reference Format:
Hui-Yin Wu, Trevor Santarra, Michael Leece, Rolando Vargas, and Arnav Jhala. 2020. Joint Attention for Automated Video Editing. In *ACM International Conference on Interactive Media Experiences (IMX '20), June 17–19, 2020, Cornella, Barcelona, Spain.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3391614.3393656

1 INTRODUCTION

Joint attention is an element of human communication where the attention of the group is drawn collectively towards focal points in an environment through non-verbal processes such as gaze, voice, and gesture[12]. Studies in film cognition have observed how continuity and analytical editing imitate "natural attention" by using gaze and audio to draw the audience's attention to elements of setting and story[4][28].

In a collaboration context, we are increasingly moving toward video conferences for meetings. The capacity to make detailed records of our daily events and the explosive growth of recorded data calls for smart methods that can understand context in videos, and automatically process and present data in a meaningful way, such as for digital archiving, or to summarize content for someone who does not have time to go through each recording. While intelligent camera switching technology is available to some extent, it is based primarily on audio, movement, and other low level features of the video streams. Existing work shows that LSTMs have been effective not only in image recognition and NLP tasks, but also for video summarization tasks [32] due to their ability to model more long ranged variable dependencies, outside of a single frame. Using LSTMs for an even more complex task such as that of video editing would be both exciting and challenging. Notably, they have been explored for predicting the head movement of users in 360 degrees VR video streaming [17]. However, automated editing methods powered by machine learning that can process, analyze, and output visual data at a higher level of context from multiple perspectives has been an ongoing challenge that is insufficiently addressed. Moreover, there is a strong lack in methods to evaluate smart conferencing technology, both from the aspect of film-editing, and user preferences [27]. Developing and understanding

such tools for communication will allow us to both improve the real-time experience of remote meeting attendees and create context aware systems that can efficiently curate large amounts of audio-visual data in real-time.

In this work, we present joint attention as a metric for automated video editing in corporate meeting recordings. We consider that each camera in the meeting room can analyze the head pose of participants occupying that video. Based on extracted head pose data and audio from individual headphones, we designed and implemented three automated editing methods: a naive audio-based edit, a rule-based edit on our joint attention metric, and an LSTM method that predicts the joint attention of the meeting at each time point. These three methods use audio data, pose data, and both respectively to produce an automated edit of meetings. Head pose and audio data, and the training of the model are pre-processed, while the final edit can be done in real time.

This work thus addresses three main challenges:

(1) conceiving a joint attention metric expressed using extracted head pose in multiple cameras and audio amplitude from individual headphones
(2) designing and implementing three automated editing methods: audio-based, rule-based, and LSTM-based method using audio and pose data to predict joint attention, trained on human expert edits
(3) evaluating each method qualitatively against the human edit, and through a user evaluation

In order to be unbiased, we selected an existing corpus of meeting videos: the AMI corpus [22], established by the University of Edinburgh, where 100 hours of meetings were recorded in smart meeting rooms equipped with multiple cameras, individual headsets, and microphone arrays, along with slide data, and post-meeting annotations.

2 RELATED WORK

Here we provide a review of existing work surrounding methodologies that can help address automated editing challenges, including attention-based interactive systems, video summarization, and smart cameras in virtual environments.

Interactive systems

Sound, motion, and object detection are common metrics that are combined with either rule-based models or learned networks such as Bayesian networks and Hidden Markov Models to perform camera selection for editing meetings or lecture videos in multi-camera environments [1, 3, 8, 20, 25]. Arev et al. [2] was the first to propose using joint attention in social cameras (i.e. cameras carried around by people participating in a group activity). Their system reconstructs the 3D environment from the video and makes cuts and transitions based on the camera orientations, which indicate the focal point of the bearer. Joint attention is inferred by the orientation of the camera held by the participants, and not from participants in the video itself.

Many editing tools and approaches also take into account cinematography conventions. Ozeki et al.[23] created an online system that generates attention-based edits of popular cooking shows by detecting speech cues and gestures. Notably Leake et al. [18] focuses on dialogue scenes and provides a quick approach that obeys a number of film idioms, selecting cameras for each line of dialogue. Their generated edits are based on annotations of the events and dialogue content. Recently, machine learning technologies have been explored for predicting the head movement of users in 360 degrees VR video [17] to improve the efficiency of streaming. The prediction of head movements and points of interest allows the use of smartly placed cinematographic techniques such as snap-cuts, virtual walls, and slow-downs to facilitate smooth playback of 360 videos [7, 26].

One notable challenge that has not been sufficiently addressed, is how different automated editing methods that are driven by human attention change the aesthetic qualities of the edit, and moreover, how they affect the perception of the audience. Though studies to evaluate the usage [27] and efficiency [7, 13, 26] of smart editing technologies exist, the evaluation of the audience's perceived preferences between different edits remains little addressed due to the fact that good editing is often invisible and should not draw the viewer's attention.

Video summarization

Video abstraction or summarization focuses on the problem of compiling a selection or thumbnails that can represent a video. Some approaches use basic visual signals [29], while Ma et al.[21] were the first to propose a attention model comprising of the audio, visual movement, and textual information. Lee et al.[19] proposed a system that generated a storyboard of daily tasks recorded on a wearable camera based on gaze, hand proximity, and frequency of appearance of an object. The output of these systems are a series of images or select meaningful segments that are representative of the video. Previous work indicates LSTMs have been effective in NLP tasks, but recently they have been used for video summarization [32] due to their ability to model more long ranged variables dependencies.

Another way of editing is by moving a smaller bounded "frame" within a fixed camera shot to create artificial camera moves focusing on the most important elements inside the shot, which is useful when adapting videos to smaller hand-held devices. Gandhi et al.[11] proposed a method to automatically generate edited clips of stage performances recorded by single cameras. Audience gaze data has also been popularly used to determine important elements in the shots,

and create a re-edit focusing on the important gaze points [14, 24]. These methods are targeted towards re-editing a still camera, adding camera movements that embody the focus of a complex performance (e.g. film or stage with lots of movement and multiple points of interest) in an artistic and compact manner.

Smart cameras for virtual environments

Autonomous 3D cameras has been a challenge on how best to place cameras in virtual environments for story and navigation. Jhala and Young designed the Darshak system which generates shot sequences that fulfill story goals to show actions and events in a 3D environment[16]. Common editing rules [10] and film editing patterns [30, 31] have also proposed to ensure spatial and temporal continuity, as well as using common film idioms. In these systems, context is crucial, and selecting a camera position from seemingly infinite options is a mathematically complex problem. Our editing task differs from virtual cameras in two main ways. First, in 3D positions of objects and the parameters of the camera are precise. In our meeting videos, visual analysis tasks are necessary to determine what the camera is showing. Camera parameters and spatial configurations of people and objects can only be approximated. Second, the virtual camera can be moved around at ease, while the cameras in the meeting room are fixed and do not follow the participants.

3 OVERVIEW

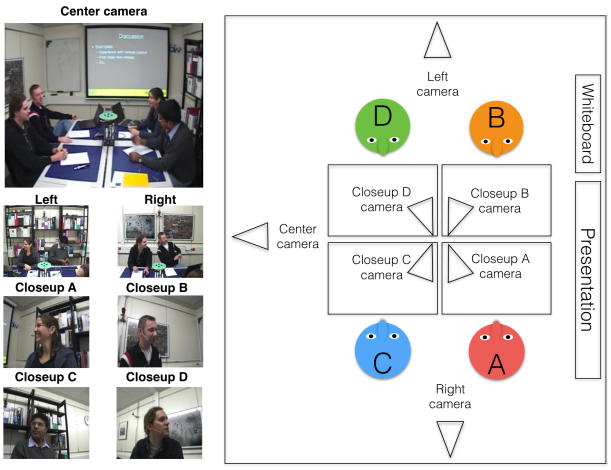

Figure 1: Configuration of the meeting room with screen captures of each camera. Each participant also has an individual microphone headset. The presentation and whiteboard is situated at the front of the room. (Screen captures from the Univ. of Edinburgh AMI corpus are reused under CC BY 4.0)

Our work consists of the design, comparison, and evaluation of three automated editing methods for multi-camera recordings in smart meeting rooms: a naive audio-based edit that serves as a baseline, a state-of-art rule-based edit using extracted head pose data, and an LSTM-based system that is trained on human expert edits to predict the joint attention score for each point of interest in the room from an input of audio and head pose data.

We generate outputs for each method from three meetings in the AMI Corpus [22] (meeting IDs IS1000a, IS1008d, and IS1009d). The three recordings represent 3 types of interactions: design and brainstorming, status update, and project planning. These meetings were held with 4 participants in the smart meeting room equipped with the seven cameras and individual microphone headsets. The configuration of the room, camera locations, and viewpoints are shown in Figure 1. The seven cameras comprise of 1 overlooking the room, 2 from the left and right, and 4 closeup cameras on each participant. Each meeting was around 30 minutes.

The next section details the joint attention metric and the processing of head pose data, which, along with audio amplitude on individual headsets, are used as features to implement the audio and rule-based edits, and to train an LSTM. To obtain the ground-truth joint attention score, a film expert edited the three meeting videos with the definition of joint attention in mind. We considered that the objects shown in the expert version of the edit were the ground truth focus of the room.

4 POSE AND JOINT ATTENTION

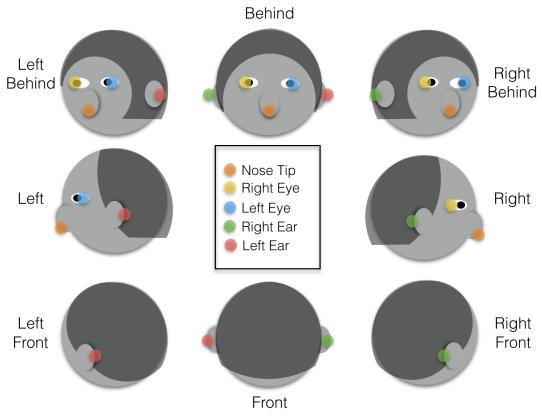

Figure 2: The eight possible head orientations and their relation to the visibility of the pose detection markers.

The meeting rooms are equipped with cameras from 7 viewpoints. Each camera's video input is analyzed to determine the head orientation of each person in its view, and then calculate the confidence level for a focal point of a person

based on an attention matrix. The focal points of all cameras are then considered together to calculate the focus of each target in the room. This is used both for the rule-based edit and training the LSTM. Below we describe the details of the head pose and focus calculation.

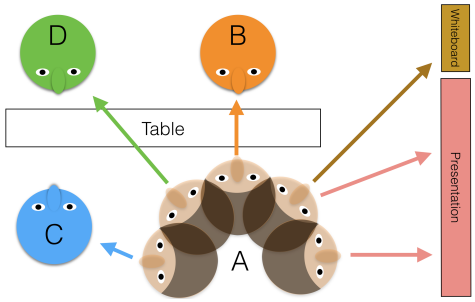

Figure 3: A person's focus is estimated based on their head pose and room configuration. Colored arrows indicate the approximate focus of Participant A based on head pose. If two or more targets are close together, there is some ambiguity as to what A is actually looking at.

Determining focal points for each camera

We used the OpenPose's MPI 15-keypoint detection to extract the head pose of participants [5], with five points on the head – 2 ears, 2 eyes, and nose tip – calculated 25 frames/second. For each point, the on-screen (x,y) coordinates are given with a confidence score between 0 and 1.

This information is assumed as input to calculate the confidence level of a head orientation, based on the visibility of the above five points on the head, shown in Figure 2. The eights head orientations are (F)ront (looking in the same direction as the camera), (B)ehind (looking towards the camera), (L)eft and (R)ight from the camera's perspective, and variations of these: LF (left-front), RF (right-front), LB (left-behind,), and RB (right-behind). Note that the left and right ear/eye of the pose detection is relative to the human, whereas the left, right, front, and behind head orientation is relative to the camera's perspective. The confidence level for a head orientation is the sum of the confidence score c of each of the n points pv that should be visible, and the sum of 1-c for the m points ph that should be hidden.

$$PoseConfidence = \sum_{i=1}^{m} c_{pv_i} + \sum_{j=1}^{n} (1 - c_{ph_j}) \quad (1)$$

The estimated head orientation is then mapped to corresponding focal points. An example is shown in Figure 3 for participant A. Each head orientation can refer to one, none, or more than one focal point. We build an focal point matrix (Table 1) that shows the mapping the head orientation and camera to corresponding focal point(s).

Table 1: The focal point matrix indicates for each camera, what the camera shows, and what focal point a participants' head orientation would correspond to. The targets in the room are the presentation (P), the whiteboard (W), and participants A, B, C, and D (A,B,C,D respectively).

Cam	Shows	L	R	LF	RF	F	LB	RB	B
Ce	(All)	BD	AC	PWB	PA	PWAB	D	C	CD
L	AC	PA	C	-	-	-	PW	D	BD
R	BD	D	PWB	-	-	-	C	A	AC
C.A	A	P	C	-	-	-	PW	D	B
C.B	B	D	PW	-	-	-	C	A	C
C.C	C	PA	-	-	-	-	PW	D	B
C.D	D	C	PWB	-	-	-	C	PWB	A

The confidence level that the focal point is a specific target (whether person or object) t for a camera C is the average of the confidence levels for the x head orientations ho that have the object or person as a focal point. This value is summed up for each person p detected in the video.

$$FocalPointConfidence(C, t) = \sum_{p=1}^{n} \frac{\sum_{o=1}^{x} c_{o,p}(t)}{x} \quad (2)$$

This gives a focal point confidence value to each target in the room from the viewpoint of C, ranking the importance of the target. Once the targets in the room are ranked by each camera, we then calculate the focal point of all participants. Focus F for a target t is the accumulated focal point confidence for each camera, since the more cameras feel that t is the focal point, the higher this score should be for t.

$$F_t = \sum_{q=1}^{7} FocalPointConfidence(C_q, t) \quad (3)$$

This is then used as an input feature for the rule-based edit, and as one of the features to train our LSTM, and generate automated edits of the meetings.

5 AUTOMATED EDITING

We can now extracted audio and focus data for automated editing. In this section we present the design and implementation of three approaches. The first is a naive audio-based edit taking into account only microphone amplitude with no criterion for pacing. The second is a state-of-the-art rule-based edit that selects cameras given a set of editing rules and focus data at a fixed pacing. The third approach is an LSTM trained to predict the joint attention of the room, coupled with a direct approach of selecting the camera that shows the joint attention of the room at each moment.

Audio-based edit

We chose the "naive" audio-based edit as a baseline, a method commonly used for existing remote conferencing solutions. The edit is generated by selecting the closeup camera of the

participant with the highest microphone input at any given time. To avoid unusual noise from the environment or the equipment, short spikes of 1 second or less are removed, and the amplitude of the four microphones are normalised to the same range, averaged over each second. The closeup camera of the person whose microphone is emitting the most sound at each second is selected.Thus minimum shot length is 1 second. No additional smoothing or pacing was added.

The main benefit of the audio-based approach is that it is fast and adaptable to all situations. No parameters were tuned, since amplitude is the only factor deciding the camera stream. However, this comes with a number of limitations, namely (1) only closeup cameras are selected with no wide view of the room nor multiple participant interactions, (2) the pacing is jittery, and (3) important reactions or movements of the participants are not observable when no sound is emitted.

Rule-based edit

Rule-based or optimization systems often use audio-visual signals for determining when to cut and which camera to choose while taking into account pacing. Our second approach uses the focal point score calculated in the previous section to design a rule-based editing algorithm that shows where the focus of the group is at a given time.

Algorithm 1 RuleEdit(FocalPointInfo F, DesiredShotLen s)

```
ShotList S; CurCam C = Center; CurShotLen s_c = 0
for all Frames f in F do
    for all Targets t do
        targetscores_t.append(f.scores_t)
    if f.num%fps == 0 then
        CamList_C.score += CutCost(s_c, s)
        for all Targets t do
            sum_t = targetscores_t.mean()
        sum.sort()
        for all i in count(Targets) do
            for all Camera x in CamList do
                if x.has(sum.at(i)) and x!=Center then
                    CamList_C.score += FPScore.at(i)
        CamList_Center.score += p
    NextCam N=CamList.max()
    if N == C then
        S.last.endtime += 1
    else
        S.append(N, f.num, f.num)
        CamList_C.score = 0
        C = N
return S
```

The algorithm first ranks each camera based on the sum of the focal point score of all the targets that it shows. Cameras portraying targets with higher focus are then rewarded points. Because the value of focus varies greatly based on the confidence of the pose detection for that frame, points are awarded by relative ranking of the target. Because this disproportionately benefits cameras that show more targets (e.g. the Center camera, showing the whole room), a fixed number of points p are allocated to Center for each step. Cameras with no people detected are assigned 0.

After the ranking the each camera for a frame, we can then make decisions on which camera to select for this frame based on the the focal points of each camera. This process is shown in Algorithm 1, involving smoothing, shot length normalization, ranking the focal point score, and selecting the best camera. To avoid jittering filter out pose data anomalies, the focal point score is smoothed over frames per second fps. The algorithm then normalizes the shot length. Assuming we have a desired shot length s seconds and current shot length s_c, the cost function $CutCost(s_c, s) = (s_c - s)^2$ is used to decide whether to stay on the current camera, or cut to a new one. Finally, the camera with the highest score is selected. If a camera change occurs, the score for the previous camera is set to 0, allowing the score of cameras that have not been selected for a long time to accumulate, ensuring that (1) all cameras will be chosen at some point, and (2) the previous camera is not selected again immediately.

The rule-based edit maintains the flexibility of setting the editing parameters such as the pacing, the weight for focus, or the weight of the Center camera. However, the weights for the various parameters such as pacing and target importance must be tuned for different situations, and due to the fixed pacing, this method is slow in adapting to quick exchanges between multiple participants. It also does not take into account audio, which if included, would add an additional parameter to be tuned.

LSTM trained joint attention

To go beyond the limitations in camera variety and pacing in the audio and rule-based methods, we implemented the third approach to directly learning the joint attention of the room at any given time point with a neural network.

Our neural network is composed of an input layer, an output layer, and a 100-neuron hidden LSTM layer, size calculated with standard equation $N_h = TrainingDataSize/2 * (N_{in} + N_{out})$ to avoid overfitting. We used MAE loss function and adam optimizer. An LSTM was chosen as opposed to fully connected layers since editing is a continuous decision-making process taking into account both rapid (e.g. who just moved or started talking) and long-term (e.g. who has been talking for the past few seconds) observations, and previous editing decisions. Input features include: (1) normalized audio amplitude of 4 individual headsets, (2) pose confidence for each participant in the four CU cameras, and (3) the focal point confidence for each of the 6 targets in the room.

To obtain a ground truth to train our model, we asked a film expert to edit our three chosen meeting videos based

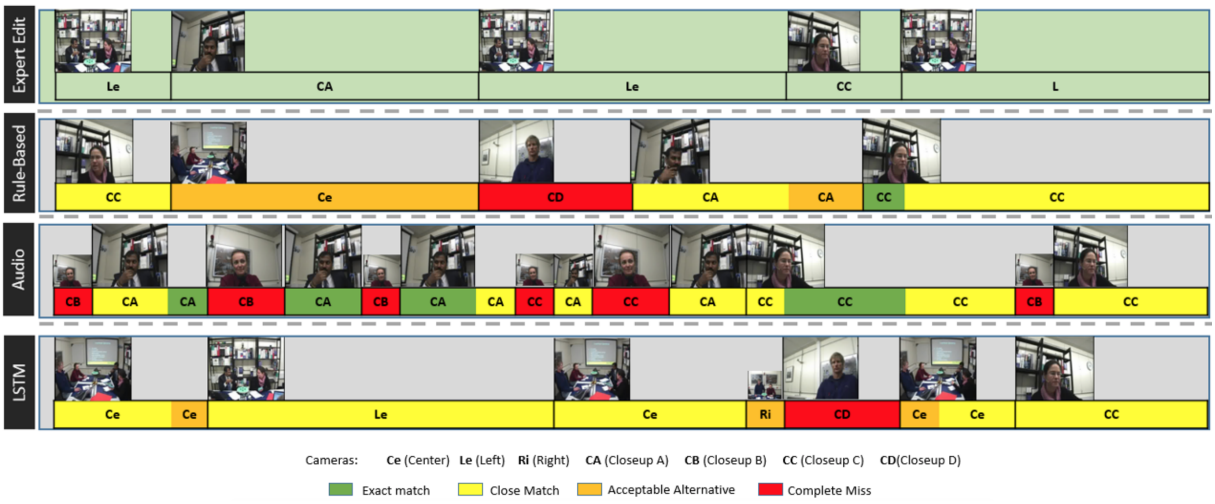

Figure 4: Here is a detailed analyses and comparison of a meeting clip. The colored timelines show how the three methods–rule, audio, and LSTM-based edits–compare to the human expert edit in terms of camera selection. The shortest shot is 1 second. The rule-based edit has smoother pacing. The audio-based edit is jittery, and makes multiple complete misses. The LSTM-based makes mostly close matches (Screen captures from the University of Edinburgh AMI corpus are reused under the Creative Commons license CC BY 4.0)

Algorithm 2 LSTMEdit(JointAttenScore J, Threshold th)

```
for all Entry t in J do
    if J[t][WB] >= 0.5 || J[t][Pr] >= 0.5 then
        CamList[t] = "Center"
    else
        for all Targets ta in J[t] do
            if ta>=0.5 then
                Candidate[t].push(ta)
        if Candidate.length == 0 then
            if J[t][A] + J[t][C] > J[t][B] + J[t][D] + th then
                CamList[t] = "Left";continue;
            if J[t][A] + J[t][C] + th < J[t][B] + J[t][D] then
                CamList[t] = "Right";continue;
            else CamList[t] = "Center"
        if Candidate.length == 1 then
            CamList[t] = Candidate[0].CloseupID
        if Candidate.length == 2 then
            if OnSameSide(Candidate[0],Candidate[1]) then
                CamList[t] = Candidate[0].MedCamID
            else CamList[t] = "Center"
        else CamList[t] = "Center"
return CamList
```

on what the editor felt was the focus of the room. Using 1 we are able to extract from the expert edit, what the expert considered was the joint attention of the room. For example, if the expert chose the *Left* camera (showing A and C) at time t, the score for A and C at time t (A_t and C_t respectively) would be set at 1.0 while the score of (B_t), D (D_t), and the whiteboard (W_t) and presentation (P_t) set at 0.0. Since the edit could be switching between two or more participants that are engaged in a dialogue, the importance of a target should not drop from 1.0 to 0.0 between shots or vice versa, so we use an exponential decay function–commonly used to represent memory–to smooth the importance of the target before and after the shot. Two meetings are used as training data over 1000 epochs, with the third as test data, cross-validating three times. The output is the a vector of joint attention scores between 0.0 and 1.0 of 6 targets in the room: four participants, whiteboard, and presentation. Higher scores imply that the target is more likely the joint attention of the room.

We designed Algorithm 2 to put together an edit based on the joint attention score. The algorithm prioritizes the "Center" camera when the whiteboard or presentation has a high joint attention score. Otherwise, it determines if a single participant solely scores over 0.5, or if two participants from the same side of the table have a significant difference in importance over the other side of the table, for which a closeup or a medium shot of two participants is shown respectively. Otherwise, the central camera is shown by default.

Initial results

Figure 4 is an example output of the audio and rule-based edits, and the edit based on LSTM-learned joint attention scores, compared side-by-side with the expert edit. Our accompanying video shows all four edits of Figure 4. Before formally comparing all methods together, there are a number of initial observations can be drawn that are not directly obvious with a more generalized analysis.

The baseline audio edit more accurately captures who is talking at each point, and in this simple scenario has the highest accuracy. However, the output is jittery and subject to noise in the data, resulting in more errors. It also chooses only closeup shots, which is not suitable for situations where multiple participants are in a discussion. The rule-based edit chooses cameras that are more or less similar to the expert, but frequently displaced in timing, which can be explained by Algorithm 1 slowly accumulating focal point confidence of each target, and the regulated pacing of around 6 seconds.

In contrast, the LSTM-joint attention method switches cameras in a more timely fashion, in many cases capturing essential actions and reactions. It also identifies scenarios with interchanges or discussions between multiple participants more accurately than the rule-based approach. This shows that the LSTM learned joint attention already produces a comparable edit to the human expert, even with the limited amount of training data and no smoothing.

6 EVALUATION

We evaluate the the various editing methods in two ways. The first is a similarity metric by comparing the rule-based, audio-based, and LSTM-based editing to the human edit. We then recruited 22 users for a study on the output videos.

Similarity metric of editing methods

Here we present a more general comparison between the three editing methods–audio-based, rule-based, and the LSTM joint attention-based–by measuring at each second in the output, the method-selected camera's dissimilarity to the expert-chosen camera. The dissimilarity is measured on four criterion of editing [9]:

Target visibility: Whether a shot shows the same targets. A camera is penalized more when it misses someone than when it shows more people than it should. For example, selecting CU.A instead of Right (showing A and C) has a higher penalty than choosing Right over CU.A.

Shot size: Whether the shot size is the same. There are three shot sizes: closeups, medium shots (Left and Right cameras), and long shots (Center). For example, there would be no shot size penalty for choosing CU.A instead of CU.B, but there would be a 2-point penalty for choosing Center instead of a CU.A.

Viewpoint: Whether the selected camera is oriented in the same direction. For example, if CU.A is chosen instead of CU.B or CU.C (who are on the same side of a line that splits the room in half), it would be penalized less than if the expert chose CU.D (which is the opposite side of the room).

Pacing: Pacing is the average shot duration in seconds. Fast pacing is generally used in more simple situations with few participants or activities, while slow pacing is used in complex situations so the viewer has time to analyse the scene.

Figure 6 shows the dissimilarity scale between a camera selected by a proposed method and the expert edit. Darker colors show a higher dissimilarity, while lighter colors show higher similarity. The highest dissimilarity would be selecting a closeup instead of the center camera, resulting in penalties on missing targets (i.e. missing 5 targets) and shot size (two levels of difference in shot size).

From the meeting videos, we randomly chose four 30-second clips to provide in-depth qualitative comparison. Figure 5 shows how the audio, rule, and LSTM-based edits compare to the expert edit for each clip. 1000a shows more general presentations and brainstorming among all participants, and the clips from 1008d and 1009d show a more discussion between two participants on a specific issue such as budget or design. The dissimilarity timeline compares at each second the difference between the expert-selected camera and the method on the combined similarity metric of target selection, size, and viewpoint. Shot size similarity only compares the difference in shot size. Pacing is measured by the average shot duration over the clip.

Generally, the LSTM-based edit is representative of the expert edit, especially in group discussion scenarios. It exceeds both the rule-based and audio edit in complex cases on shot size and pacing. However, the audio-based edit has higher accuracy than either of the other two methods where there is close discussion between two participants. This is because the two participants are looking at each other while the other participants may be taking their own notes, referring to the slides, or watching the discussion without too much activity. This analysis also does not take into account displacement in timing, which is common in the rule-based edit Figure 4. Also, the audio-based edit has a much more jittery pacing as compared to the other methods, with average shot length of less than 2 seconds, which would result in an uncomfortable viewing experience.

User evaluation

Here we detail the design and results of our user evaluation on the output of the various editing methods.

Survey design. The study consisted of a 30-minute sitting in two parts: a quiz for data receivability, and nine pairwise comparisons between videos using different editing methods.

For the "receivability" quiz, the user was shown three 30-second clips, one from each meeting, followed by two to three basic questions concerning the content. All clips only showed the Center camera view from Figure 1. At the end, users were asked to select from the three meetings the one they enjoyed most/least, and the one they were most/least willing to participate via video conference. The goal was to

		Shot dissimilarity	Size Dissimilarity	Pacing (Sec/shot)	Avg. Dissimilarity
1000a-1 (10 sec/shot)	Audio			1.2	4.69
	Rule			**6**	3.04
	LSTM			2.3	**2.66**
1000a-2 (4.3 sec/shot)	Audio			3	2.95
	Rule			6	**2.20**
	LSTM			**2.7**	2.93
1008d-1 (6 sec/shot)	Audio			2.3	**1.51**
	Rule			**6**	2.45
	LSTM			3.8	1.63
1009d-1 (6 sec/shot)	Audio			1.8	**1.19**
	Rule			**7.5**	2.08
	LSTM			3	2.17

Figure 5: We compare the output of the the audio, rule, and the LSTM-based edits to the expert edit for four 30-second clips on aspects of overall similarity throughout the clip, similarity of shot size, and average shot duration. The average shot duration of the expert edit is shown under the meeting ID. The shot and size dissimilarity timeline show the difference between the expert chosen camera using the scale in Figure 6. Average dissimilarity is measured based on shot size difference, missing/extra targets shown, and viewpoint, with a value between 0 (exact match) and 5 (complete miss).

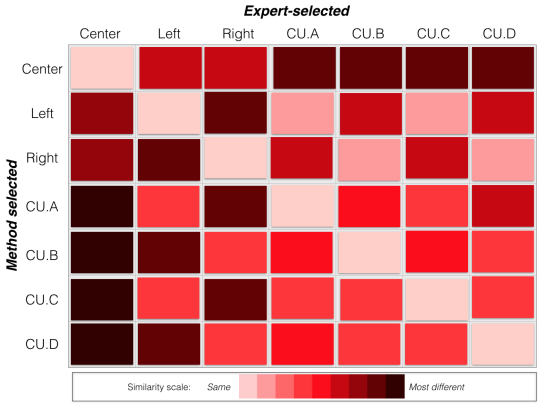

Figure 6: The similarity between two cameras is calculated based on targets visible and shot size. This figure shows the scale of dissimilarity between the camera selected by the method and the expert selected camera, used in Figure 5 to represent distance of chosen cameras from the expert chosen ones. $CU.X$ is a closeup on participant X.

(1) ensure that the data is receivable by checking that the users were concentrated on the tasks, and (2) identify strong preferences between the different meetings.

The second part of the study consisted of nine pairs of video clips, which users viewed, and selected the one that was easier to follow, and the one they preferred. Nine 30 second clips, three from each meeting, were randomly selected, each edited with the four different editing methods, creating a pool of 36 videos. Four versions of 9 pairwise comparisons questions (total of 36 distinct pairs) were generated. Clip selection and pairings take inspiration from the Youden squares design [15], with equal distribution and combinations of meeting-edit pairs across the 36 questions. Each pair of video clips evaluated by the user always shows content from two different meetings and editing methods. This was specifically avoid drawing attention to the editing method itself and overlooking the overall appreciation of the video. The users were not told that the videos used different editing methods, and asked to judge the videos solely on their feelings. Since the four versions of pairwise comparisons were randomly assigned, some meeting/editing pairs in Tables 2, 3, and 4 had more results than others.

Pilot study. To validate our experimental design, we first conducted a pilot study with only pairwise comparisons with clips from the human expert and audio edits. Users from the pilot study did not overlap those from the formal study. Only one clip per meeting was selected, and the video pairs could feature the same editing method, though any pair of clips were still from different meetings. Users just selected the video clip that they preferred. A total of 432 pairwise comparisons were recorded.

Results, shown in Table 2, demonstrate that the editing method strongly influences user preferences, and found a significant preference of human-edited videos over audio-edited ones ($p=0.0069$).

Table 2: This table summarizes results of the pilot study, for each editing method of each meeting clip, the number of times it was preferred in a pairwise comparison (in total), and the number of times it was preferred over the opposite editing method.

Meeting ID		Video preferred		Method preferred	
Video	Edit	Count	Percentage	Count	Percentage
1000a	Human	82 / 134	61.2%	44 / 59	74.6%
1000a	Audio	51 / 127	40.2%	25 / 73	34.2%
1008d	Human	61 / 146	41.8%	37 / 71	52.1%
1008d	Audio	53 / 147	36.1%	19 / 69	27.5%
1009d	Human	85 / 139	61.2%	45 / 71	63.4%
1009d	Audio	100 / 171	58.5%	45 / 87	51.7%

Final study. We recruited 22 users to evaluate the outputs of the various editing methods.

In the first phase of the evaluation, all the data was deemed receivable, with maximum two questions wrong out of seven. We found that users had a stronger preference for meeting 1000a on both enjoyability and willingness to participate, and least for meeting 1009d, shown in Figure 7. Due to this bias, in the second part of the study we also analyse pairwise results between meetings, and the preference of editing methods within meetings.

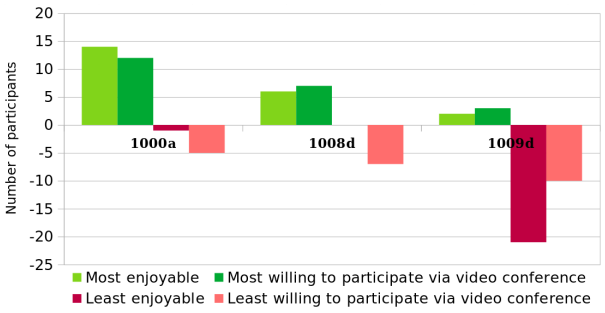

Figure 7: This figure shows the number of users who selected each meeting to be the most (in green, positive numbers) and least (in red, negative numbers) enjoyable and the meeting they are most/least willing to participate in through video conference. Users preferred meeting 1000a most on both metrics, and meeting 1009d least on both metrics. We pay particular attention to this for the second part of the study to remove the bias that the meeting content would influence the video preference.

A total of 198 pairwise comparisons were observed in the second part. Table 3 summarizes the number of pairwise comparisons for any two editing methods.

Overall, results show that users felt uniformly that the human edit was both easier to follow and more preferred when compared to any other automated editing method. However, in both preference and ease to follow, human, LSTM-based and audio-based edits had no statistical significant difference. In contrast, though both LSTM-based and rule-based edit performed less well than human, users still ranked LSTM over rule. The greatest difference between these two methods comes to when they are compared against the audio-based edit. While the audio-based edit significantly defeats the rule-based edit, LSTM-based edit is ranked higher than audio-based edit.

Table 3: A total of 198 pairwise comparisons were observed, summarised two metrics: *Ease to follow* and *Preference*. The values $Method1-Method2$ indicate respectively the number of times Method 1 is preferred over Method 2, and the opposite. The method more preferred for each pair is highlighted in bold, and asterisk indicate p-value < 0.05.

Methods	No. pairs	Ease to follow	Preference
Human - Audio	34	18 - 16	21 - 13
Human - Rule	35	23* - 12	26* - 9
Human - LSTM	32	20 - 12	21 - 11
Audio - Rule	32	28* - 4	25* - 7
Audio - **LSTM**	34	15 - 19	15 - 19
Rule - **LSTM**	31	12 - 19	14 - 17

Table 4: Pairwise preferences between meeting combinations in the same format as Table 3. No meeting performs uniformly better or worse than all other meetings.

Meetings	No. pairs	Ease to follow	Preference
1000a - 1008d	65	30 - 35	25 - 40
1008d - 1009d	65	28 - 37	29 - 36
1009d - 1000a	68	35 - 33	25 - 43

Since each pair of videos showed different meetings, to rule out bias due to the content, we analyse pairwise preference between meetings. Table 4 regroups information from Table 3 shows pairwise how each combination of meetings performed. No meeting performed uniformly better or worse than all other meetings, and results here do not reflect the meeting preferences from the first part of the study (Table 7).

As mentioned, no pairwise comparisons had two videos from the same meeting in order to avoid users from focusing on the editing instead of the viewing experience. However, when reorganizing statistics from Table 3 to observe the ranking of editing methods within the same meeting (Figure 8), we see that the chance a video is considered easier to follow or preferable is significantly linked to the editing method, and is similar across all meetings. This comparison echoes the findings in Table 3 showing that the human edit was selected over all other methods on both metrics, and the rule-based edit was the least preferred.

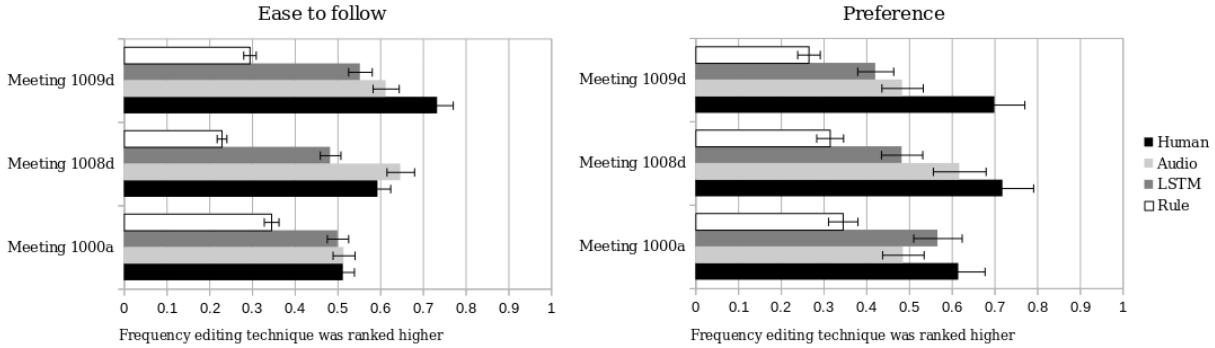

Figure 8: We regroup data from Table 3 to show the frequency an editing method is preferred within the same meetings. Though no pairwise comparisons consisted of video clips from the same meeting, results here show that the editing method had a strong influence over the chance that one video would considered easier to follow or preferable over another.

7 DISCUSSION

James E. Cutting et al. [6] argue that there is a strong correlation between the speed of cuts, and viewers' attention and expectations. An observer expects to see a constant balance between visual composition, cinematic rules and editing pace that matches the audiovisual material. Finding this balance would be the key to designing an optimal editing system.

One prospect of this work was to develop real-time film editing systems, and study the perception of users of these systems. While LSTMs shows high potential with viewers over conventional rule and naive audio-based systems, and can be trained offline, pose data extraction is currently a bottleneck. Libraries with high accuracy such as OpenPose require longer computation, while those that can perform in real-time have trouble detecting faces that are not directly looking at the camera. Another challenge was the source material: the low video and audio quality was significantly limiting when extracting head pose data. Also, due to the difficulty of obtaining expert edits, the training of the LSTMs operated on a more limited data size, and results would be influenced on which videos were used as training data. This remains a big challenge when designing learning models for applications such as film editing that have room for creativity, and for which data is difficult to obtain. Thus the use of LSTMs for real-time editing is still constrained.

Finally, evaluation was an enormous challenge, since film editing is considered most successful when it doesn't draw attention, and little work has been done on how to evaluate outputs of automated editing systems. Beyond basic criterion of pacing or shot similarity, it is difficult to evaluate the quality of the output. In our user evaluation, we could see that the editing technique did have a strong impact on whether a video was considered easy to follow or preferable, and that for different meeting content, different editing techniques could be more or less suited.

From these first promising findings, we intend to address these limitations by (1) expanding the training dataset with expert edits from different film editors, to observe global use of joint attention in editing decisions, and (2) conducting user studies at a larger scale to observe statistical significance between perception of different editing treatments.

8 CONCLUSION

We introduced the use of a joint-attention metric from cognitive psychology to design and implement automated editing techniques that allow us to have an edit on the level of attention of participants occupying and interacting within the same space. It has also proven to be a lightweight metric for machine learning models, as demonstrated with an LSTM, bringing more variety to viewpoints while maintaining focus on important elements, and with minimal tuning. This is further highlighted from the comparison of various editing methods against the expert edit. Our user evaluation shows that the editing method strongly influences the viewing experience, and that users do prefer edits that show multiple viewpoints using indices of attention.

Our vision is for such attention-based automated editing techniques to be used in live event streaming for better remote viewing experience. We would improve our current approaches by learning transition parameters across multiple elements in the scene to generate edits that can determine higher level contexts–such as presentation, group discussion, etc.–and make edits according to audio-visual elements that are more essential to these contexts.

ACKNOWLEDGEMENTS

We thank Oliver Pell, Konrad Michaels, Kent Foster, and Facebook for providing initial feedback and support for this work. Parts of this work were supported by Baskin School of Engineering at UC Santa Cruz and the Visual Narrative faculty cluster at North Carolina State University.

REFERENCES

[1] Marc Al-hames, Benedikt Hörnler, Ronald Müller, Joachim Schenk, Gerhard Rigoll, and Technische Universität München. 2007. Automatic multi-modal meeting camera selection for video-conferences and meeting browsing. In *Proceedings of the 8th International Conference on Multimedia and Expo (ICME*. IEEE Xplore, Beijing, 2074–2077.

[2] Ido Arev, Hyun Soo Park, Yaser Sheikh, Jessica Hodgins, and Ariel Shamir. 2014. Automatic Editing of Footage from Multiple Social Cameras. *ACM Trans. Graph.* 33, 4 (July 2014), 81:1–81:11. https://doi.org/10.1145/2601097.2601198

[3] Floraine Berthouzoz, Wilmot Li, and Maneesh Agrawala. 2012. Tools for Placing Cuts and Transitions in Interview Video. *ACM Trans. Graph.* 31, 4 (July 2012), 67:1–67:8.

[4] David Bordwell. 1985. *Narrative in the Fiction Film.* University of Wisconsin Press, USA.

[5] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Xplore, Honolulu, Hawaii, USA, 7291–7299.

[6] James E. Cutting, Jordan E. DeLong, and Christine E. Nothelfer. 2010. Attention and the Evolution of Hollywood Film. *Psychological Science* 21, 3 (2010), 432–39.

[7] Savino Dambra, Giuseppe Samela, Lucile Sassatelli, Romaric Pighetti, Ramon Aparicio-Pardo, and Anne-Marie Pinna-Déry. 2018. Film Editing: New Levers to Improve VR Streaming. In *Proceedings of the 9th ACM Multimedia Systems Conference* (Amsterdam, Netherlands) *(MMSys '18)*. ACM, New York, NY, USA, 27–39. https://doi.org/10.1145/3204949.3204962

[8] Fahad Daniyal and Andrea Cavallaro. 2011. Multi-Camera Scheduling for Video Production. In *Proceedings of the 2011 Conference for Visual Media Production (CVMP '11)*. IEEE Computer Society, USA, 11–20. https://doi.org/10.1109/CVMP.2011.8

[9] Quentin Galvane, Rémi Ronfard, and Marc Christie. 2015. Comparing film-editing. In *Eurographics Workshop on Intelligent Cinematography and Editing, WICED '15* . Eurographics Association, Zurich, Switzerland, 5–12. https://doi.org/10.2312/wiced.20151072

[10] Quentin Galvane, Rémi Ronfard, Christophe Lino, and Marc Christie. 2015. Continuity Editing for 3D Animation. In *AAAI Conference on Artificial Intelligence*. AAAI Press, Austin, Texas, United States, 753–761. https://hal.inria.fr/hal-01088561

[11] Vineet Gandhi, Remi Ronfard, and Michael Gleicher. 2014. Multi-clip Video Editing from a Single Viewpoint. In *Proceedings of the 11th European Conference on Visual Media Production* (London, United Kingdom) *(CVMP '14)*. ACM, New York, NY, USA, Article 9, 10 pages. https://doi.org/10.1145/2668904.2668936

[12] Jane Heal. 2005. *Joint Attention: Communication and Other Minds: Issues in Philosophy and Psychology.* Oxford University Press, USA.

[13] Pan Hu, Rakesh Misra, and Sachin Katti. 2019. Dejavu: Enhancing Videoconferencing with Prior Knowledge. In *Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications* (Santa Cruz, CA, USA) *(HotMobile '19)*. Association for Computing Machinery, New York, NY, USA, 63–68. https://doi.org/10.1145/3301293.3302373

[14] Eakta Jain, Yaser Sheikh, Ariel Shamir, and Jessica Hodgins. 2014. Gaze-driven Video Re-editing. *ACM Transactions on Graphics* 34-2, Article 21 (2014), 12 pages.

[15] Arnav Jhala and R. Michael Young. 2009. Comparing Effects of Different Cinematic Visualization Strategies on Viewer Comprehension. In *Interactive Storytelling*, Ido A. Iurgel, Nelson Zagalo, and Paolo Petta (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 26–37.

[16] Arnav Jhala and R. Michael Young. 2011. *Intelligent Machinima Generation for Visual Storytelling.* Springer New York, New York, NY, 151–170. https://doi.org/10.1007/978-1-4419-8188-2_7

[17] Evgeny Kuzyakov, Shannon Chen, and Renbin Peng. 2017. Enhancing high-resolution 360 streaming with view prediction. https://engineering.fb.com/virtual-reality/enhancing-high-resolution-360-streaming-with-view-prediction/

[18] Mackenzie Leake, Abe Davis, Anh Truong, and Maneesh Agrawala. 2017. Computational Video Editing for Dialogue-Driven Scenes. In *ACM Transactions on Graphics*, Vol. 36. Association for Computing Machinery, New York, NY, USA, Article Article 130, 14 pages. Issue 4. https://doi.org/10.1145/3072959.3073653

[19] Yong Jae Lee and Kristen Grauman. 2015. Predicting Important Objects for Egocentric Video Summarization. *International Journal of Computer Vision* 114, 1 (01 Aug 2015), 38–55. https://doi.org/10.1007/s11263-014-0794-5

[20] Qiong Liu, Yong Rui, Anoop Gupta, and J. J. Cadiz. 2001. Automating Camera Management for Lecture Room Environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Seattle, Washington, USA) *(CHI '01)*. Association for Computing Machinery, New York, NY, USA, 442–449. https://doi.org/10.1145/365024.365310

[21] Yu-Fei Ma, Lie Lu, Hong-Jiang Zhang, and Mingjing Li. 2002. A User Attention Model for Video Summarization. In *Proceedings of the Tenth ACM International Conference on Multimedia* (Juan-les-Pins, France) *(MULTIMEDIA '02)*. ACM, New York, NY, USA, 533–542. https://doi.org/10.1145/641007.641116

[22] Iain Mccowan, J Carletta, Wessel Kraaij, Simone Ashby, S Bourban, M Flynn, M Guillemot, Thomas Hain, J Kadlec, Vasilis Karaiskos, M Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska Masson, Wilfried Post, D Reidsma, and P Wellner. 2005. The AMI meeting corpus. In *International Conference on Methods and Techniques in Behavioral Research*. Wageningen: Noldus Information Technology, Wageningen, Netherlands, 702 pp.

[23] M. Ozeki, Y. Nakamura, and Y. Ohta. 2004. Video editing based on behaviors-for-attention - an approach to professional editing using a simple scheme. In *2004 IEEE International Conference on Multimedia and Expo (ICME)*, Vol. 3. IEEE Computer Society, Taipei, Taiwan, 2215–2218 Vol.3. https://doi.org/10.1109/ICME.2004.1394710

[24] Kranthi Kumar Rachavarapu, Moneish Kumar, Vineet Gandhi, and Ramanathan Subramanian. 2018. Watch to Edit: Video Retargeting using Gaze. *Computer Graphics Forum* 37, 2 (2018), 205–215. https://doi.org/10.1111/cgf.13354 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13354

[25] Abhishek Ranjan, Jeremy Birnholtz, and Ravin Balakrishnan. 2008. Improving Meeting Capture by Applying Television Production Principles with Audio and Motion Detection. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) *(CHI '08)*. ACM, New York, NY, USA, 227–236.

[26] Lucile Sassatelli, Marco Winckler, Thomas Fisichella, Antoine Dezarnaud, Julien Lemaire, Ramon Aparicio-Pardo, and Daniela Trevisan. 2020. New interactive strategies for virtual reality streaming in degraded context of use. *Computers & Graphics* 86 (2020), 27 – 41. https://doi.org/10.1016/j.cag.2019.10.005

[27] David A. Shamma, Jennifer Marlow, and Laurent Denoue. 2019. Interacting with Smart Consumer Cameras: Exploring Gesture, Voice, and AI Control in Video Streaming. In *Proceedings of the 2019 ACM International Conference on Interactive Experiences for TV and Online Video* (Salford (Manchester), United Kingdom) *(TVX '19)*. Association for Computing Machinery, New York, NY, USA, 137–144. https://doi.org/10.1145/3317697.3323359

[28] Tim J. Smith. 2012. The Attentional Theory of Cinematic Continuity. *Projections* 6, 1 (2012), 1–27.

[29] Ba Tu Truong and Svetha Venkatesh. 2007. Video Abstraction: A Systematic Review and Classification. *ACM Transactions on Multimedia Computation Communication Applications* 3, 1, Article 3 (Feb. 2007), 37 pages. https://doi.org/10.1145/1198302.1198305

[30] Hui-Yin Wu and Marc Christie. 2015. Stylistic Patterns for Generating Cinematographic Sequences. In *4th Workshop on Intelligent Cinematography and Editing Co-Located w/ Eurographics 2015*. Eurographics Association, Zurich, Switzerland, 47–53. https://doi.org/10.2312/wiced.20151077 The definitive version is available at http://diglib.eg.org/.

[31] Hui-Yin Wu, Francesca Palù, Roberto Ranon, and Marc Christie. 2018. Thinking Like a Director: Film Editing Patterns for Virtual Cinematographic Storytelling. *ACM Transactions on Multimedia Computing, Communications and Applications* 14, 4 (2018), 1–23. https://doi.org/10.1145/3241057

[32] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. 2016. Video Summarization with Long Short-Term Memory. In *Computer Vision – ECCV 2016*, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (Eds.). Springer International Publishing, Cham, 766–782.